Combinatorial Constructions for Sifting Primes and Enumerating the Rationals

Edinah K. Gnang * Chetan Tonde †

In memory of Herbert Wilf

January 13, 2012

Abstract

We describe a combinatorial approach for investigating properties of rational numbers. The overall approach rests on structural bijections between rational numbers and familiar combinatorial objects, namely rooted trees. We emphasize that such mappings achieve much more than enumeration of rooted trees.
We discuss two related structural bijections. The first corresponds to a bijective map between integers and rooted trees. The first bijection also suggests a new algorithm for sifting primes. The second bijection extends the first one in order to map rational numbers to a family of rooted trees. The second bijection suggests a new combinatorial construction for generating reduced rational numbers, thereby producing refinements of the output of the Wilf-Calkin[1] Algorithm.

1 The Combinatorics

The word "*Combinatorics*" here mostly refers to the combinatorics of *trees*, more specifically labeled *rooted trees* see Fig[1]. The defining property of a tree

*Department of Computer Science, Rutgers University, Piscataway, NJ 08854-8019 USA
†Department of Computer Science, Rutgers University, Piscataway, NJ 08854-8019 USA

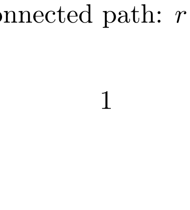

Figure 1: Typical example of a rooted tree. Root node: r, leaves: $\{v_1, v_3, v_5\}$, (parent,child) pair: (v_2, v_3), connected path: $r \to v_2 \to v_4 \to v_5$.

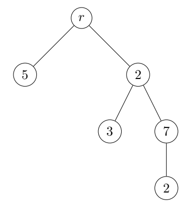

Figure 2: A valid label assignment with labels = 2,3,5 and 7.

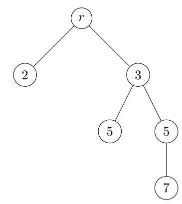

Figure 3: An invalid label assignement with labels = 2,3,5 and 7.

is that precisely one path connects any two vertices of the tree. At various parts of the discussion we will assign labels to the vertices of the trees. *Rooted* trees have each a special vertex (typically labeled r) referred to as the *root* of the tree. Moreover a collection of rooted trees is called a *planted forest.* For convenience in the subsequent discussion the word "*tree*" will refer to a rooted tree while the word "*forest*" will refer to a planted forest.
Each vertex $v \neq \mathit{root}$, is adjacent to a single vertex whose distance to the root is smaller than the distance between v and the root. (The distance between two vertices here refers to the number of edges on the unique path connecting the two vertices.) Such a vertex is called the *parent* of v and all other vertices adjacent to v are called *children* vertices of v.
The analogy to human family relations is extended to include notions such as *siblings* and *grandparent* relationships between vertices. We assume that such relationships between vertices are unambiguous within the current context of trees. Vertices with no *children* are called *leaves* of the tree.
Finally, a label assignment to the vertices of a given tree is considered valid, if the root is labeled r and no two sibling vertices are assigned the same label. See Fig[2,3].

1.1 Operations on trees and forests

1.1.1 Grafting trees and forests

The first operation on trees that we introduce here is the *grafting* operation.
Let T_α and T_β denote two rooted trees. We say that the tree T_γ results from the grafting of T_α onto T_β denoted

$$T_\gamma = T_\alpha \wedge T_\beta \tag{1}$$

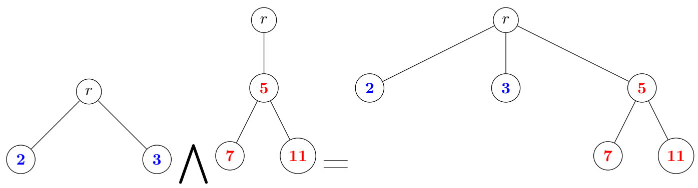

Figure 4: Grafting operation on trees.

iff T_α and T_β are disjoint rooted subtrees of T_γ and most importantly T_γ has no other rooted subtree disjoint from the subtrees T_α and T_β as illustrated in Fig[4].
Note that, it follows from the definition that the grafting operation is commutative

$$T_\alpha \;\wedge\; T_\beta = T_\beta \;\wedge\; T_\alpha. \tag{2}$$

The single–vertex rooted tree (having only the root vertex), can be thought to represent the neutral (or identity) element for the grafting operation. We write

$$root \;\wedge\; root = root \tag{3}$$

and

$$T_\alpha \;\wedge\; root = root \;\wedge\; T_\alpha = T_\alpha \tag{4}$$

The tree grafting operation induces a natural algebra on forests described below. Let $\mathcal{F} = \{T_k\}_{1\leq k\leq n}$ and $\mathcal{G} = \{T'_l\}_{1\leq l\leq m}$ denote two forests. The grafting of the forest $\mathcal{F}$ onto the forest $\mathcal{G}$ amounts to creating a new forest made of trees resulting from the grafting of all possible pairs of trees (T_u, T'_v) such that

$$(T_u, T'_v) \in \mathcal{F} \times \mathcal{G}. \tag{5}$$

Hence,

$$\mathcal{F} \wedge \mathcal{G} = \{T_k \wedge T'_l\} \;\; 1 \leq k \leq n, 1 \leq l \leq m. \tag{6}$$

1.1.2 Raising forests

Let $\mathcal{F} = \{T_k\}_{1\leq k\leq n}$ denote a forest and T_α denote a tree. The operation of raising the forest $\mathcal{F}$ by the tree T_α noted

$$\mathcal{R}\,(T_\alpha, \mathcal{F}) \tag{7}$$

consists in substituting every tree T_k in the forest $\mathcal{F}$ with an extended tree T'_k constructed by rooting the tree T_k at every leaf of T_α as illustrated in Fig[5].
We note that the raising operator is not commutative. It also follows from the definition that for any tree T_α we have

$$\mathcal{R}\,(root, T_\alpha) = root. \tag{8}$$

Finally, for convenience we adopt the convention that

$$\mathcal{R}\,(T_\alpha, root) = T_\alpha. \tag{9}$$

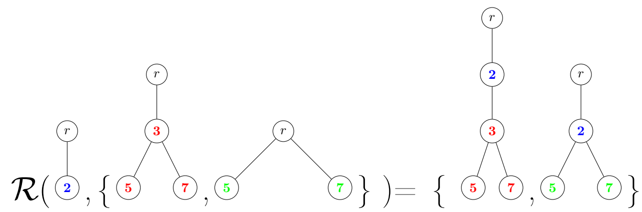

Figure 5: Raising operation on forests.

1.2 Generating the forest of all labeled trees of height bounded by h using n labels.

We recall that a label assignment to vertices of a tree is considered valid if the root is unlabeled and no two sibling vertices of the tree are assigned the same label. We assume here that the label set is of size n, and furthermore, if we restrict our attention to the trees of height bounded by a positive integer h, it is clear that the set of such trees is finite.
We now describe a construction which produces the set of trees described above as a forest. For this purpose we consider the following recursion.

$$G_0 = \{root\} \tag{10}$$

and

$$G_i = \bigwedge_{0 \le k \le n-1} \{root,\ \mathcal{R}\left(T_k, G_{i-1}\right)\}, \tag{11}$$

that is,

$$G_i = \{root,\ \mathcal{R}\left(T_0, G_{i-1}\right)\} \wedge \cdots \wedge \{root,\ \mathcal{R}\left(T_{n-1}, G_{i-1}\right)\} \tag{12}$$

where T_k denotes the tree made of the root and an additional vertex who is assigned the k^{th} label from our labeling set $L = \{0, 1, \cdots, (n-1)\}$.
For illustration purposes we express G_1

$$G_1 = \bigwedge_{0 \le k \le n-1} \{root,\ T_k\}. \tag{13}$$

$$= \{root,\ T_0\} \wedge \cdots \wedge \{root,\ T_{n-1}\}. \tag{14}$$

By construction we have that for $1 \le i \le h$ the trees in the forest G_i are distinct, validly labeled and of depth bounded above by i. Furthermore

$$G_s \subsetneq G_{s+1} \tag{15}$$

It also follows from the definition of the recurrence that G_1 has 2^n elements and that the number of trees in the forest G_i is prescribed by the recurrence relation

$$|G_i| = \left(1 + |G_{i-1}|\right)^n \tag{16}$$

Theorem 1: The forest G_h contains all trees with valid vertex label assignment and of height bounded by h.

Proof : The theorem is proved by first observing that the validly labeled trees of height bounded by h correspond to rooted subtrees of the complete n-ary tree of height h. A simple counting argument reveals that the number of such trees is determined by the recurrence relation

$$S_0 = 1 \tag{17}$$

and

$$\forall\, 1 \le i \le h,\ S_i = (1 + S_{i-1})^n \tag{18}$$

where S_i denote the number of distinct rooted subtrees of the complete n-ary trees with height bounded by i. We therefore conclude the proof by observing that

$$\forall\, 0 \le i \le h,\ |G_i| = S_i. \square \tag{19}$$

2 Applications to Number Theory

We discuss here the structural bijection between the set of integers and the set of valid prime–labeled trees. The mapping of an arbitrary valid tree (labeled with primes) to an integer, is established by equating the sibling relationship between vertices to integer multiplication and also equating for non-root vertices the parenthood relation to integer exponentiation. The process is therefore algorithmic and will be referred to as evaluation of a tree to an integer. We may point out that the evaluation process is recursive and insist that the evaluation be always initiated at the leaves because of the non-associativity of integer exponentiation.

On the other hand the mapping of an arbitrary integer to a valid prime–labeled tree immediately follows from recursively applying the fundamental theorem of arithmetics to the powers of the prime factors. The single vertex tree (only having the root as a vertex) is associated to the integer 1. It therefore follows that the mapping between trees and integers is bijective.

2.1 Combinatorial Prime Sieve Algorithm

Modern sieve theory focuses on providing accurate estimates for the number of primes in some integer interval. In contrast, as an application of the combinatorial framework described above we discuss a variation of the classical sieve of Eratosthenes. We recall some notation convention.

$[n]$ denote the set of consecutive integers from 1 to n,

$\mathbb{P}$ denotes the set of prime numbers

$\mathbb{N}$ denotes the set of strictly positive integers.

Algorithm *Combinatorial Prime Sieve*
Input: For $q \in \mathbb{P}$ the set $\mathbb{P} \cap [q]$
Output: The set $\mathbb{P} \cap ([2q] \backslash [q])$

$G_{out} \leftarrow \{\emptyset\}$(*Output list*)
$G_{rest} \leftarrow \{\emptyset\}$(*restricted list of composites*)
$G_{old} \leftarrow \{\emptyset\}$(*old extended list of integers*)
$G_{new} \leftarrow \bigwedge_{0 \le k \le n-1} \{root, T_k\}$ (*new extended list of integers*)
while $(\exists T \in (G_{new} \backslash G_{old})\ s.t.\ evaluation(T) \le 2q\ )$
  do $G_{old} \leftarrow G_{new}$
    $G_{new} \leftarrow \bigwedge_{0 \le k \le n-1} (root, \mathcal{R}(T_k, G_{old}))$

    for $i \in [1, \dots, |(G_{new} \backslash G_{old})|]$
      do
        if $(q < evaluation((G_{new} \backslash G_{old})[i]) < 2q)$ *and* $((G_{new} \backslash G_{old})[i] \notin G_{rest})$
          (*found a new composite*)
          then $G_{rest}.orderedInsert\{(G_{new} \backslash G_{old})[i]\}$ (*added the new composite to restricted the list*)
    for $i \in [1, \dots, |G_{rest}| - 1]$
      do
        if $(evaluation(G_{rest}[i+1]) == evaluation(G_{rest}[i]) + 2)$ (*found a new prime*)
          then $G_{out} \leftarrow G_{out}.Append\{evaluation(G_{rest}[i]+1)\}$ (*add it to the list*)

    return G_{out}

Theorem 2.1. *$\forall q \in \mathbb{P}$ the algorithm above will produce all the primes in the range $(q, 2q)$*

Proof. The correctness of the algorithm follows from the well established fact that for all $n \ge 1$, there exist some prime number p with $n < p \le 2n$ [2], in addition to two other facts. The first of which is that in the range $(q, 2q)$ there is no composite whose corresponding tree has a vertex labeled with a prime greater than q. The reason being that $2q$ is the largest of the composite less than or equal to $2q$, in other words the smallest composite admiting a prime p greater than q as a vertex label in the associated tree would be of the form $2p$ and consequently necessarily greater than $2q$. The second fact is that all the composites whose corresponding trees are labeled with primes less than or equal to q are determined by the combinatorial algorithm described in section 1.2. Furthermore, the sought–after primes in the range $(q, 2q)$ will be uncovered by identifying trees in the resulting forest which evaluate to nearest composites whose difference equals 2. $\square$

2.2 Re-enumerating the rationals

Georg Cantor was the first to establish the suprising fact that the set $\mathbb{Q}^+$ is countably infinite. In [1] Neil Calkin and Herbert Wilf introduced a sequence

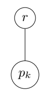

Figure 6: tree T_k.

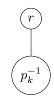

Figure 7: tree $T_{k^{-1}}$.

which listed the elements of $\mathbb{Q}^+$ so as not to include duplicates in the sequence. We discuss here another sequence for listing the rationals so as not to include duplicates. Our sequence follows from the integer combinatorial encoding described above.

The proposed construction for listing elements of $\mathbb{Q}^+$ has the benefit of providing explicit control over the subset of the prime numbers used to express the rationals in the sequence.
Let T_k denote the tree made of a root vertex and an additional vertex labeled with the k^{th} prime (see Fig[6]), and let $T_{k^{-1}}$ denote the tree made of a root vertex and an additional vertex labeled with the inverse of the k^{th} prime (see Fig[7]). Building on the recurrence for G_i discussed in 1.2 we write the recurrence

$$G_0 = \{root\} \tag{20}$$

and

$$G_i = \bigwedge_{k\in\mathbb{N}} \{root,\ \mathcal{R}\left(T_k, G_{i-1}\right)\}. \tag{21}$$

We use the recurrence for G_i to create a new H_i recurrence defined by

$$H_i = \bigwedge_{k\in\mathbb{N}} \{\mathcal{R}\left(T_{k^{-1}},\ G_i\right), root, \mathcal{R}\left(T_k, G_i\right)\}. \tag{22}$$

We also have

$$H_i \subsetneq H_{i+1}. \tag{23}$$

The trees in the forest H_i evaluate to distinct rational numbers in reduced form and the rooted trees in the forest H_i all have depth less than or equal to i. The trees which evaluate to non-integers have special vertices that are prime–inverse labeled and attached to the root. Furthermore, it follows from the fact that siblings have different labels that corresponding rational numbers are in their reduced form.
Hence as a corollary of *Theorem*1 the trees in forest H_∞ bijectively maps to $\mathbb{Q}^+$.

3 Acknowledgments

The authors are grateful to Professor Doron Zeilberger, Professor Henry Cohn, Professor Mario Szegedy, Professor Vladimir Retakh and Jules Lambert for insightful discussions and suggestions. The authors are greatful to Jeanine Sedjro for proof reading the manuscript. The first author was partially supported by the National Science Foundation grant NSF-DGE-0549115 and Microsoft Research New England.

References

[1] N. Calkin & H. Wilf, Recounting the rationals. *Amer. Math. Monthly*, 107:pp.360–363, 2000.

[2] M. Aigner;G. Ziegler (2009). Proofs from THE BOOK (4th ed.). Berlin, New York: Springer-Verlag. ISBN 978-3-642-00855-9.

[3] Edinah K. Gnang, Experimental Number Theory, Part I : Tower Arithmetic (Preprint) arXiv:1101.3026v1 [math.NT]